Underspecification without Underspecified Representations

Scott Nelson
Stony Brook University
Department of Linguistics
Institute for Advanced Computational Science
scott.nelson@stonybrook.edu

Eric Baković
UC San Diego
Linguistics Department
ebakovic@ucsd.edu

1 Introduction

Phonological knowledge can be broken down into REPRESENTATIONAL KNOWLEDGE (*what is the data structure for phonological elements?*) and COMPUTATIONAL KNOWLEDGE (*what types of operations are computed over phonological elements?*). Featural underspecification is one area where these types of knowledge directly interact. Theoretical analyses involving underspecification rely on certain types of phonological elements not being valued (= 'specified') for all features in order to capture a phonological generalization.

Formalizing this understanding of underspecification is not straightforward and leads to issues of both over-generalization and under-generalization, depending on the representational encoding and computational evaluation language (Nelson, 2022). Here we pursue an alternative hypothesis, that underspecified representations are epiphenomenal and that the types of phonological processes where they are regularly employed implicate only computational knowledge. We formalize our argument using Boolean Monadic Recursive Schemes, a logical characterization of the subsequential functions (Bhaskar et al., 2020) and a general-use formalism for phonological analysis (Chandlee and Jardine, 2021). Specifically, we argue that a shared computational structure is used when defining the types of phonological maps that have been analyzed with underspecification and this structure is independent of sub-segmental representational choices.

2 Phonological Underspecification

Given a set of features Φ, a phonological element x is said to be UNDERSPECIFIED if there exists a feature $\varphi \in \Phi$ such that x is unvalued for φ. Typically, features are valued using $\{+, -\}$, indicating the presence vs. absence of some property. An element x is typically unvalued for a given feature either because the property it corresponds to is not relevant to a higher-order class of sounds to which x belongs, or because the property is not *contrastive* within the higher-order class of sounds.

In Russian, for example, only obstruents — sounds specified as $[-\texttt{sonorant}]$ — contrast in terms of voicing, and must thus be specified as either $[-\texttt{voice}]$ or $[+\texttt{voice}]$. Sonorants, specified as $[+\texttt{sonorant}]$, do not contrast in terms of voicing; they are only $[+\texttt{voice}]$. Obstruents also participate as both triggers and targets of a voicing assimilation process, but sonorants do not. One way to analyze this pattern is to say that voicing assimilation is general, in principle involving obstruents and sonorants alike, but that the lack of a voicing contrast for sonorants means that they are not valued for the $[\texttt{voice}]$ feature and therefore do not participate as either targets or triggers.

Under the common assumption that surface structures must be fully specified, this use of underspecification requires an additional set of REDUNDANCY MAPPINGS in addition to whatever standard set of mappings are used to account for the various processes in a language. These are rules of the form $X \rightarrow Y$, where X is a feature bundle describing a class of segments and Y is a set of features to be 'filled in' on X (e.g. via priority union; Reiss 2022). While these are interpreted as rules, they also provide an implication structure. For example, in the Russian example we have a redundancy rule $[+\texttt{sonorant}] \rightarrow [+\texttt{voice}]$.

Underspecified representations therefore provide an account for why certain phonological elements do not trigger or target certain processes. Another way to put this is that underspecification removes elements from the domain of a phonological function. Phonological processes like $a \rightarrow b \ / \ c\_d$ can be formalized as functions that map the string cad to the string cbd. Here, a corresponds to the target and c_d corresponds to the trigger, both of which make up the domain of the function and are restricted by underspecified representations.

Proceedings of the Society for Computation in Linguistics (SCiL) 2024, pages 352-356.
Irvine, California, June 27-29, 2024

Phonological functions ultimately describe output strings through the computation of individual feature valuations. In the case of Russian voicing assimilation, the valuation of the [voice] feature for a given obstruent in a sequence of obstruents is determined by the valuation of [voice] for the following obstruent. When an obstruent is followed by a sonorant, the output value for [voice] is identical to whatever it was in the input. At the same time, the redundancy rule ensures that the output valuation of [voice] for any sonorant in the input is positive. These two functions can be combined to describe a single computation for determining the output value for [voice] for the elements that make up any Russian input word.

Based on this description, computing the value for [voice] is broken down into two yes-no questions and interpreted as a decision tree (Figure 1). First, is the phonological element x for which the valuation for [voice] is in question a sonorant? If yes, then it is [+voice]. If no, is the element following x a sonorant? If yes, then x keeps whatever value for [voice] it already has. If no, then it takes on the value for [voice] of the following element.

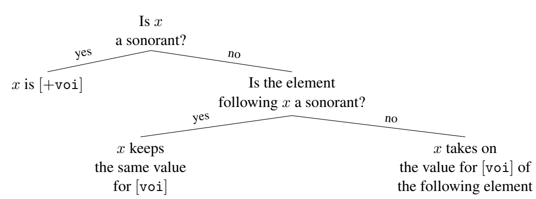

Figure 1: Decision tree for determining the valuation of [voice] for a phonological element x in Russian.

These decision tree questions never once require the input [voice] valuation of a sonorant element to determine the output [voice] valuation for any element, thus making sonorant voicing irrelevant. Additionally, the first question removes sonorants from the set of targets while the second question removes sonorants from the set of triggers, thus accomplishing the same extensional goal as using underspecified representations without actually requiring that representations be underspecified.

In the remaining sections, we formalize this insight following recent advances in computational phonology. The decision tree structure can be implemented with IF...THEN...ELSE syntax which is found in many programming languages and is a central aspect of the Boolean Monadic Recursive Schemes formalism. This formalism has been proposed as a way to wed theoretical and computational approaches to phonological generalizations (Chandlee and Jardine, 2021) and is therefore situated perfectly to highlight our claim that what looks like underspecification is instead an emergent property of certain computational structures.

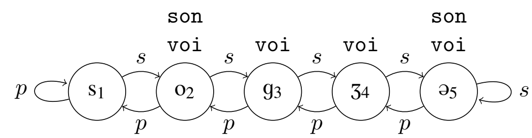

Figure 2: Model for Russian [sogʒə] 'juice (emph.)'

3 Boolean Monadic Recursive Schemes

Boolean Monadic Recursive Schemes (BMRS) are programs that operate over model-theoretic structures. A structure/model for a string $\mathbf{S} = \langle D, \sigma_i \mid \sigma \in \Sigma, \texttt{p}(), \texttt{s}()\rangle$ includes a set of indices D, unary labeling relations $\sigma_i \subseteq D$, and the predecessor $\texttt{p}()$ and successor $\texttt{s}()$ ordering functions. For phonological purposes Σ is viewed as a set of n feature predicates $\{f_1, \dots, f_n\}$ that can be interpreted as $[+f]$ when evaluated to $\top$ (= True) and $[-f]$ when evaluated to $\bot$ (= False). A graphical interpretation of the model for Russian [sogʒə] 'juice (emph.)' is shown in Figure 2. IPA symbols are shown next to the domain label for reference and a subset of feature properties (voice, sonorant) are listed above domain elements where that property holds. For example, domain elements 2 and 5 represent vowel sounds and therefore have the property of being both [+voice] and [+sonorant].

A BMRS term T is given by the grammar $T \rightarrow x \mid T_1 = T_2 \mid \top \mid \bot \mid \mathbf{f}(T_1, ..., T_k) \mid \texttt{s}(T_1) \mid \texttt{p}(T_1) \mid \sigma(T_1) \mid$ IF T_1 THEN T_2 ELSE T_3. BMRS terms are evaluated in relation to a specific structure $\mathbf{S}$. Each term has a type that it inherits inductively. x has type index which ranges over the domain of the structure being evaluated. $\top$ and $\bot$ have type Boolean, evaluating to True and False, respectively. The equality operator requires both terms to have the same type and returns something of type Boolean. Functions and relations are required to operate only over type index and return something of type Boolean. T_1 must be of type Boolean in the IF T_1 THEN T_2 ELSE T_3 sequence while T_2 and T_3 must be of the same type.

A BMRS term with n IF...THEN...ELSE nestings is an n-NESTED CONDITIONAL. The standard binary Boolean connectives AND ($\wedge$), OR ($\vee$), and IF ($\Rightarrow$) are all 1-nested conditional BMRS terms and for any n-ary logical connective, there is an

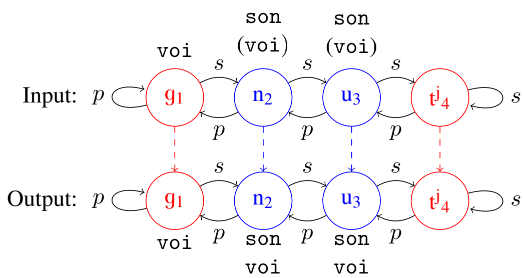

Figure 3: Mapping for /gnutʲ/ → [gnutʲ] 'to bend'. Color coding indicates the part of the IF...THEN...ELSE statement in (1) used to compute output voicing.

equivalent $(n-1)$-nested conditional BMRS term.

A BMRS program consists of functions $\phi_i(x_1, ..., x_n) = T_i$ that determine the truth value of a given term for each element of the domain, and can be interpreted as a string-to-string transduction from input structure $\mathbf{S}$ to output structure $\mathbf{T}$. The program defines the output properties of $\mathbf{T}$ in terms of the input properties of $\mathbf{S}$. This allows for declarative descriptions of phonological generalizations. The Russian redundancy rule above can be interpreted as a function in a BMRS program:

(1) $\phi_{\texttt{voi}}(x) :=$ IF son(x) THEN $\top$ ELSE voi(x)

This is interpreted as "domain element x has the property of being [+voice] in the output if it is a sonorant, otherwise it is specified for whatever value it had in the input," which is precisely the generalization the redundancy rule aims to capture.

Figure 3 shows both an input and output structure that corresponds to the input-output mapping described in (1). The input structure uses parentheses around voi to indicate optional presence or absence of this property. Both the equation and the models have been color coded to show the relationship between the conditions and the output. The blue IF statement checks to see whether a domain element has the property of being a sonorant in the input and if so, then that domain element evalutes to True for the voi predicate and therefore has the property of being [+voice] in the output. Computing the redundancy rule thus acts as a non-violable constraint that enforces all sonorants to be voiced in the output structure. Therefore, the actual input specification for the voicing feature on sonorants (positive, negative, or underspecified) does not matter. We extend this insight to the full Russian assimilation map in the next section to show how underspecification-like patterns can emerge from computational structure and are in fact agnostic to the representational encoding scheme used.

4 Underspecification as Computation

We refer to segmental phonological maps that have been analyzed using underspecified representations as UNDERSPECIFICATION MAPS. Our primary claim is that underspecification maps share the same computational structure regardless of representational encoding. The following conditions describe the structure of an underspecification map.

(2) a. The map will define input-output conditions for the "underspecified feature".

b. Any underspecification map will include a 1-nested conditional BMRS term.[1]

c. Both the upper conditional P and lower conditional Q will determine a truth value based on the antecedent of the redundancy rule that fills in the "underspecified feature".

d. P partitions the set of targets while Q partitions the set of triggers.

The relationship between (2a) and (2c) above is crucial to the classification of this type of map as an underspecification map. It reframes the conditions on phonological processes from being dependent on the lack of valuation for the underspecified feature directly to being dependent on the valuation for the feature on which the underspecified feature is dependent — in the case of Russian, to being about [sonorant] rather than about [voice].

Using the example of Russian regressive obstruent voicing assimilation, we show that the general structure of the map maintains these properties regardless of whether underspecified representations are used. In a fully specified representation scheme, voi() is a unary relation evaluating to $\{\top, \bot\}$ for a given index, such that positive valuations of a feature evaluate to $\top$ and negative valuations evaluate to $\bot$. Chandlee and Jardine (2021) suggest that BMRS can be adapted to account for underspecified features by loosening the Boolean requirement for functions/relations and changing voi() in the model signature to a function that maps domain elements to $\{+, -, 0\}$. In all future descriptions we continue to use only the binary Boolean feature relations but point out that even if we use the underspecification-equipped feature functions, the structures of our BMRS programs do not change.

The following BMRS function determines the voicing properties of output elements in Russian.

[1]The lower conditional requires conjunction when discussing a two-sided triggering environment (which compiles out to a 1-nested conditional BMRS term). This is a technical point that has minimal bearing on the larger point being made.

(3) $\phi_{\texttt{voi}}(x) :=$ (Russian voicing function)

$$\begin{array}{l} \texttt{IF son}(x) \texttt{ THEN } \top \\ \quad \texttt{ELSE IF son}(\texttt{s}(x)) \texttt{ THEN voi}(x) \\ \qquad \texttt{ELSE voi}(\texttt{s}(x)) \end{array}$$

This term never calls the input function voi() on a domain element that could be "underspecified." The conditional calls partition both the set of potential targets (P) and the set of potential triggers (Q), thus removing sonorants from both domains of application and unifying the redundancy and main rules of Russian voicing. Furthermore, the structure of the BRMS function completely matches the decision tree generalization from Section 2.

Consider now Catalan. Sonorant consonants are not targets for regressive voicing assimilation, as in Russian, but unlike Russian they *do* serve as assimilation triggers and therefore must be specified as [+voice]. Because of this, there is a disconnect between the set of segments that act as targets, partioned using [sonorant], and the set of segments that act as triggers, partioned using [syllabic].

(4) $\phi_{\texttt{voi}}(x) :=$ (Catalan voicing function)

$$\begin{array}{l} \texttt{IF son}(x) \texttt{ THEN } \top \\ \quad \texttt{ELSE IF syll}(\texttt{s}(x)) \texttt{ THEN voi}(x) \\ \qquad \texttt{ELSE voi}(\texttt{s}(x)) \end{array}$$

This computation does not satisfy (2c) and is therefore not an underspecification map. Changing the upper conditional to $\texttt{syll}(x)$ would result in a computational structure satisfying (2) but would erroneously include sonorant consonants in the set of targets and therefore incorrectly describe the Catalan assimilation map. Likewise, changing the lower conditional to $\texttt{son}(\texttt{s}(x))$ would remove sonorant consonants from the set of triggers and once again describe an incorrect map for Catalan.[2]

5 Diacritic Underspecification

Inkelas et al. (1997) consider another, distinct use of underspecification in their analysis of voicing in Turkish, in which three relevant classes of morphemes: those that end (a) in non-alternating voiceless stops, (b) in non-alternating voiced stops, and (c) in stops that alternate between voiceless (in codas) and voiced (elsewhere). The authors propose that the final stops of morphemes in each of these classes are represented as in (5), with strictly feature-filling processes handling the eventual valuations of the alternating stops in class (c).

(5) a. [−voice]: [devlet̠] ∼ [devlet̠i] 'state ∼ ACC'
  b. [+voice]: [etyd̠] ∼ [etyd̠y] 'study ∼ ACC'
  c. [0 voice]: [kanat̠] ∼ [kanad̠ɯ] 'wing ∼ ACC'

The strictly feature-filling nature of the processes required by this analysis is a challenge for our approach. However, note that underspecification in this case is essentially being used as a lexical class diacritic.[3] Suppose instead that the final stops of class (b) are underlyingly specified with a diacritic feature $[+f]$. The relevant facts can then be captured with the following BMRS function.

(6) $\phi_{\texttt{voi}}(x) :=$ (Turkish voicing function)

$$\begin{array}{l} \texttt{IF } f(x) \vee \texttt{son}(x) \texttt{ THEN } \top \\ \quad \texttt{ELSE IF coda}(x) \wedge \texttt{stop}(x) \texttt{ THEN } \bot \\ \qquad \texttt{ELSE voi}(x) \end{array}$$

The upper conditional ensures that morpheme-final stops in class (b), and sonorants generally, always surface as [+voice], making the underlying voicing value for class (b) irrelevant. The lower conditional is the standard coda devoicing function.

This is essentially the 'co-phonology' approach considered (and rejected) by Inkelas et al., whereby classes (a) and (c) result from a standard coda devoicing grammar while class (b) results from a grammar without devoicing. One reason they reject this approach is the existence of morphemes with voiced internal coda stops but alternating final stops, e.g. [edʒdaːt̠] ∼ [edʒdaːd̠ɯ] 'ancestry ∼ ACC'. The problem is that co-phonologies apply to entire morphemes, while underspecification can be selectively applied to individual elements of morphemes. A second reason is due to a worry about an overgenerating proliferation of co-phonologies. Our single BMRS function avoids both these issues.

6 Conclusion

We have shown that underspecification can be viewed as a purely computational property of phonological maps. Our approach highlights a difference between standard uses of underspecification (e.g. Russian) and diacritic uses (e.g. Turkish). In future work we intend to expand the conditions outlined in (2) to better capture this distinction.

[2] One aspect left for future work has to do with clarifying the role of {sub,sup}er set relations. The function in (4) doesn't require [+ syllabic] elements to have a voicing specification since they are removed from the set of targets by being a subset of [+ sonorant] in the upper conditional and then removed from the set of triggers in the lower conditional.

[3] Indeed, the clearly exceptional [+voice] class (b) consists mostly of loans like [etyd] 'study' and [katalog] 'catalog'.

Acknowledgements

We would like to thank Jeff Heinz for suggesting the title and three anonymous SCiL reviewers for providing helpful feedback on this work.

References

Siddharth Bhaskar, Jane Chandlee, Adam Jardine, and Christopher Oakden. 2020. Boolean monadic recursive schemes as a logical characterization of the subsequential functions. In *Language and Automata Theory and Applications: 14th International Conference, LATA 2020, Milan, Italy, March 4–6, 2020, Proceedings 14*, pages 157–169. Springer.

Jane Chandlee and Adam Jardine. 2021. Computational universals in linguistic theory: Using recursive programs for phonological analysis. *Language*, 97(3):485–519.

Sharon Inkelas, Orhan Orgun, and Cheryl Zoll. 1997. The implications of lexical exceptions for the nature of grammar. In Iggy Roca, editor, *Derivations and Constraints in Phonology*, pages 393–418. Oxford University Press.

Scott Nelson. 2022. A model theoretic perspective on phonological feature systems. *Proceedings of the Society for Computation in Linguistics*, 5(1):1–10.

Charles Reiss. 2022. Priority union and feature logic in phonology. *Linguistic Inquiry*, 53(1):199–209.